Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по лабораторной работе №9

на тему

«Пользовательские LINQ»

по дисциплине  
**«Визуальное программирование»**

Выполнили:

Студенты группы №140

Бригада 5

Сафаров Д. А.

Тимохин Е. С.

Проверили:  
ст. преп. Хизриева Н.И.

ст. преп. Бастрычкин А.С.

**Цель работы**

Реализовать пользовательские LINQ запросы на языке C#.

**Задание**

Реализовать 5 собственных методов расширения для коллекций.

Использовать разные дженерики (TSource, TResult, TKey, TValue).

Как минимум в одном методе передать в качестве аргумента Func.

Не использовать уже существующие LINQ.

Продемонстрировать работу на примерах коллекций различных типов.

**Практическая часть**

Код программы представлен в приложении А. Результат её работы представлен на рисунке 1.



Рисунок 1 – Результаты работы программы

Приложение А. Текст программы

**MyExtensions.cs**

namespace LinqExtensionsLogic

{

public static class MyExtensions

{

// Проверка, удовлетворяют ли все элементы условию (предикату)

public static bool MyAll<TSource>

(

this IEnumerable<TSource> source,

Func<TSource, bool> predicate

)

{

foreach (var item in source)

if (predicate(item) == false)

return false;

return true;

}

// Возвращает коллекцию – пересечение двух коллекций по ключам

public static IEnumerable<TSource> MyIntersect<TSource, TKey>

(

this IEnumerable<TSource> first,

IEnumerable<TSource> second,

Func<TSource, TKey> keyExtractor

)

{

var result = new List<TSource>();

foreach (var item1 in first)

foreach (var item2 in second)

if (keyExtractor(item2).Equals(keyExtractor(item1)))

result.Add(item1);

return result;

}

// Применяет функцию ко всем элементам коллекции и возвращает коллекцию с

результатами

public static IEnumerable<TResult> MyDoThing<TSource, TResult>

(

this IEnumerable<TSource> source,

Func<TSource, TResult> thing

)

{

var result = new List<TResult>();

foreach (var item in source)

result.Add(thing(item));

return result;

}

// Исключает повторяющиеся элементы с одним ключом

public static IEnumerable<TSource> MyDistinct<TSource, Tkey>

(

this IEnumerable<TSource> source,

Func<TSource, Tkey> keyExtractor

)

{

var result = new List<TSource>();

foreach (var item1 in source)

{

bool isAdded = false;

foreach (var item2 in result)

if (keyExtractor(item2).Equals(keyExtractor(item1)))

isAdded = true;

if (!isAdded)

result.Add(item1);

}

return result;

}

// Возвращает словарь из переданной коллекции

public static Dictionary<TKey, TValue> MyToDictionary<TValue, TKey>

(

this IEnumerable<TValue> source,

Func<TValue, TKey> keyExtractor

) where TKey : notnull

{

var result = new Dictionary<TKey, TValue>();

foreach (var item in source)

if (result.TryAdd(keyExtractor(item), item) == false)

throw new Exception($"Key '{keyExtractor(item)}' is not unique!");

return result;

}

}

}

**Program.cs**

using ClassLibrary;

using LinqExtensionsLogic;

/// Пример на int

var listOfInt = new List<int>([3, -5, 12, 3, -7, 5]);

Console.WriteLine("\t Коллекция чисел");

foreach (var item in listOfInt)

Console.Write(" " + item);

Console.WriteLine("\n\nВсе числа больше 0 ?\n > " + listOfInt.MyAll(x => x > 0));

listOfInt = listOfInt.MyDoThing(Math.Abs).ToList();

Console.WriteLine("\n Взятие модулей");

foreach (var item in listOfInt)

Console.Write(" " + item);

Console.WriteLine("\n\n Исключение повторов");

listOfInt = listOfInt.MyDistinct(x => x).ToList();

foreach (var item in listOfInt)

Console.Write(" " + item);

Console.WriteLine("\n\n Пересечение с [-5, 0, 5]");

listOfInt = listOfInt.MyIntersect([-5, 0, 5], x => x).ToList();

foreach (var item in listOfInt)

Console.Write(" " + item);

/// Пример на Item

var listOfItem = new List<Item>();

listOfItem.Add(new Item(1, "Футболка", "Синий", 15000.99, "Модный магазин", new DateOnly(2023, 5, 10)));

listOfItem.Add(new Item(2, "Джинсы", "Черный", 29.99, "Джинсовый мир", new DateOnly(2022, 4, 22)));

listOfItem.Add(new Item(4, "Рубашка", "Красный", 24.99, "Модный магазин", new DateOnly(2023, 6, 5)));

Console.WriteLine("\n\n\t Коллекция товаров\n");

foreach (var item in listOfItem)

Console.WriteLine(item);

Console.WriteLine("\t Перекраска всего в чёрный (осенняя коллекция)\n");

listOfItem = listOfItem.MyDoThing(item => new Item(item.Id, item.Name, "Чёрный",

item.Price\*1.5, item.Provider,

item.ProductionDate)).ToList();

foreach (var item in listOfItem)

Console.WriteLine(item);

Console.WriteLine("Все товары не раньше 2022 года производства?\n > " + listOfItem.MyAll(item => item.ProductionDate.Year >= 2022));

var dictionaryOfItem = listOfItem.MyToDictionary(item => item.Id);

Console.WriteLine("\n\t Словарь из коллекции товаров\n");

foreach (var item in dictionaryOfItem)

Console.WriteLine($" Ключ: {item.Key}\t| Значение: {item.Value}");